CLAIMS

What is claimed is:

1. A method comprising:

receiving a set of data, the set of data having a first number of subsets;

defining a compression group corresponding to the set of data, the compression group having a plurality of entries, each entry containing a pointer to a corresponding one of the subsets;

compressing the set of data so that the set of data occupies a smaller number of the subsets than the first number; and

for each of the subsets which does not contain compressed data after said compressing, storing a predetermined value in the corresponding entry of the compression group, the predetermined value being indicative that corresponding data is compressed.

2. A method as recited in claim 1, wherein the predetermined value further is indicative that the corresponding compressed data is represented in a different entry of the compression group.

3. A method as recited in claim 1, wherein the predetermined value further is indicative of the compression algorithm used to compress the data.

4. A method as recited in claim 1, wherein the set of data is a portion of a file, and each of the subsets of the set of data is a separate block within said portion of the file.

5. A method as recited in claim 4, wherein said method is performed in response to a request to write the file; and

wherein the method further comprises writing the portion of the file to a non-volatile storage device after said compressing.

6. A method as recited in claim 5, wherein said writing the portion of the file to the non-volatile storage device is performed after said compressing but before any other portion of the file is received by the data storage system.

7. A method as recited in claim 4, wherein the compression group is a portion of an indirect node of the file.

8. A method as recited in claim 4, wherein the compression group is a portion of an inode node of the file.

9. A method as recited in claim 1, further comprising:

saving an uncompressed version of the portion of the set of data in a memory in the data storage system after said compressing; and

in response to a subsequent request on the set of data, using the uncompressed version of the data from the memory to fulfill the request.

10. A method as recited in claim 1, further comprising;

receiving a read request;

in response to the read request, determining that the read request relates to at least one subset of the set of data;

scanning the compression group to determine whether any entry in the compression group contains the predetermined value; and

upon detecting the predetermined value in any of the entries in the compression group, immediately beginning decompression of the set of data.

11. A method as recited in claim 1, wherein the method is performed in a data storage system configured to perform data mirroring, and wherein said compressing is performed at a consistency point.

12. A method as recited in claim 1, wherein the method is performed in a data storage system configured to perform data mirroring, and wherein the method further comprises, at a consistency point:

scanning the compression group to determine whether the set of data has been compressed; and

determining whether any of the subsets in the set of data has been modified since a prior consistency point; and

upon determining that the set of data has been compressed and that at least one of the subsets of the set of data has been modified, sending the set of data in its entirety to a remote data storage system at a mirror site, for use in a mirror copy of the file.

13. A method comprising:

receiving a file containing data; and

compressing at least part of the file to form a plurality of compression groups, each of the compression groups representing less than the entire file, each of the compression groups corresponding to an independently compressible group of data.

14. A method as recited in claim 13, wherein each of the compression groups represents a plurality of blocks of the file.

15. A method as recited in claim 13, wherein said compressing at least part of the file comprises compressing the data represented by each of the compression groups independently.

16. A method as recited in claim 15, further comprising determining suitability for compression independently for each of the compression groups.

17. A method as recited in claim 13, wherein each of the compression groups contains a plurality of pointers.

18. A method as recited in claim 17, wherein at least one of the pointers in each of the compression groups points to compressed data, and wherein at least one other pointer in each of the compression groups is a predetermined value indicative that corresponding data has been compressed.

19. A method as recited in claim 18, wherein the predetermined value further is indicative of the compression algorithm used to compress the data.

20. A method of compressing data in a data storage system, the method comprising:

receiving a file at the data storage system, a portion of the file including a number of consecutive blocks of uncompressed data,

defining a compression group to represent the portion of the file, including defining the compression group to have a plurality of entries and filling each of the entries with a block number that points to a corresponding one of the blocks;

determining whether the portion of the file is suitable for compression; and

if the portion of the file is determined to be suitable for compression, then

compressing the portion of the file so that the portion occupies a smaller number of consecutive blocks, and

for each of the number of consecutive blocks which does not contain compressed data after said compressing, storing a predetermined block number in the corresponding entry of the compression group, the predetermined block number being indicative that corresponding data is compressed and represented elsewhere in the compression group.

21. A method as recited in claim 20, further comprising repeating said defining a compression group so as to define a plurality of compression groups to represent the file.

22. A method as recited in claim 21, wherein each compression group represents a portion of an indirect node of the file.

23. A method as recited in claim 21, each compression group represents a portion of an inode node of the file.

24. A method as recited in claim 20, wherein the predetermined block number further is indicative of the compression algorithm used to compress the data.

25. A method as recited in claim 20, wherein the file comprises a plurality of portions, each including a plurality of blocks of data, and wherein the method further comprises repeating said defining, determining, compressing, and storing, for each of the plurality of portions.

26. A method as recited in claim 20, further comprising, in response to a read request, determining that the portion of the file is compressed by scanning the compression group for the predetermined block number.

27. A method as recited in claim 20, wherein said method is performed in response to a request to write the file; and

wherein the method further comprises writing the portion of the file to a non-volatile storage device after said compressing.

28. A method as recited in claim 27, wherein said writing the portion of the file to the non-volatile storage device is performed after said compressing but before any other portion of the file is received by the data storage system.

29. A method as recited in claim 27, further comprising:

saving an uncompressed version of the portion of the file in a memory in the data storage system after said compressing; and

in response to a subsequent request on the portion of the file, using the uncompressed version from the memory to fulfill the request, without decompressing the compressed portion of the file.

30. A method as recited in claim 20, further comprising;

receiving a read request at the data storage system;

in response to the read request, determining that the read request relates to at least one block of the portion of the file;

scanning the compression group to determine whether any entry in the compression group contains the predetermined block number; and

upon detecting the predetermined block number in any of the entries in the compression group, immediately beginning decompression of the portion of the file.

31. A method as recited in claim 20, wherein the data storage system is configured to perform data mirroring, and wherein said compressing is performed at a consistency point.

32. A method as recited in claim 20, wherein the data storage system is configured to perform data mirroring, the method further comprising, at a mirroring event:

scanning the compression group to determine whether the portion of the file has been compressed; and

determining whether any block in the portion of the file has been modified since a prior mirroring event; and

upon determining that the portion of the file has been compressed and that at least one block in the portion of the file has been modified, sending the portion in its

entirety to a remote data storage system at a mirror site, for use in a mirror copy of the file.

33. A method as recited in claim 20, wherein consecutive entries in the compression group correspond to consecutive blocks in the file.

34. A method of compressing data in a data storage system, the method comprising:

receiving a request to write a file at the data storage system;

in response to the request, identifying a plurality of portions of the file, each portion including a number of consecutive blocks of uncompressed data;

defining a separate compression group to represent each of the portions, so as to define a plurality of compression groups to represent the file, including defining each compression group to include a plurality of entries, wherein each of the entries is filled with a block number that points to a corresponding one of the blocks, wherein consecutive entries in the compression group correspond to consecutive blocks in the file;

determining whether each of the portions of the file is suitable for compression;

for each portion, if the portion is determined to be suitable for compression,

compressing the portion into a smaller number of consecutive blocks, and

for each block which does not contain compressed data after said compressing, storing a predetermined block number in the corresponding entry of the compression group, the predetermined block number being indicative that corresponding data is compressed and represented elsewhere in the compression group; and

writing the file to a non-volatile storage device after said compressing.

35. A method as recited in claim 34, wherein the predetermined block number further is indicative of the compression algorithm used to compress the data.

36. A method as recited in claim 34, wherein each of the compression groups represents a portion of an indirect node of the file.

37. A method as recited in claim 34, wherein each of the compression groups represents a portion of an inode node of the file.

38. A method as recited in claim 34, further comprising, in response to a read request, determining that a portion of the file is compressed by scanning the corresponding compression group for the predetermined block number.

39. A storage server comprising:

a processor;

a network communication interface to provide the data storage server with data communication with a plurality of clients over a network;

a storage interface to provide the data storage server with data communication with a set of mass storage devices; and

a memory containing code which, when executed by the processor, causes the data storage server to execute a process of managing data in the mass storage devices on behalf of the clients, the process comprising

receiving a set of data, the set of data having a first number of subsets,

creating a compression group corresponding to the set of data, the compression group having a plurality of entries, each entry containing a pointer to a corresponding one of the subsets,

compressing the set of data so that the set of data occupies a smaller number of the subsets than the first number, and

for each of the subsets which does not contain compressed data after said compressing, storing a predetermined value in the corresponding entry of the compression group, the predetermined value being indicative that corresponding data is compressed.

40. A storage server as recited in claim 39, wherein said process of managing data is performed by a file system layer of the data storage server.

41. A storage server as recited in claim 39, the predetermined value further being indicative that the corresponding compressed data is represented in a different entry of the compression group.

42. A storage server as recited in claim 39, wherein the predetermined value further is indicative of the compression algorithm used to compress the data.

43. A storage server as recited in claim 39, wherein the set of data is a portion of a file, and wherein each of the subsets of the set of data is a separate block within said portion of the file.

44. A storage server as recited in claim 43, wherein the process of storing data in the mass storage devices is performed in response to a request to write the file from one of the clients; and

wherein the process further comprises writing the portion of the file to a non-volatile storage device after said compressing.

45. A storage server as recited in claim 44, wherein said writing the portion of the file to the non-volatile storage device is performed after said compressing but before any other portion of the file is received by the data storage system.

46. A storage server as recited in claim 43, wherein the compression group represents a portion of an indirect node of the file.

47. A storage server as recited in claim 43, wherein the compression group represents a portion of an inode node of the file.

48. A storage server as recited in claim 39, wherein the memory further contains code which, when executed by the processor, causes the data storage server to execute a process of causing data stored in the mass storage devices to be mirrored at a remote site, said process comprising:

saving an uncompressed version of the portion of the set of data in a memory in the data storage system after said compressing; and

in response to a subsequent read on the portion of the set of data, using the uncompressed version from the memory to fulfill the request, without decompressing the portion of the set of data.

49. A storage server as recited in claim 39, wherein the memory further contains code which, when executed by the processor, causes the data storage server to execute a process of causing data stored in the mass storage devices to be mirrored at a remote site, said process comprising:

receiving a read request;

in response to the read request, determining that the read request relates to at least one subset of the set of data;

scanning the compression group to determine whether any entry in the compression group contains the predetermined value; and

upon detecting the predetermined value in any of the entries in the compression group, immediately beginning decompression of the set of data.

50. A storage server as recited in claim 39, wherein the data storage system configured to perform data mirroring, and wherein said compressing is performed at a consistency point.

51. A storage server as recited in claim 39, wherein the method is performed in a data storage system configured to perform data mirroring, and wherein the method further comprises, at a mirroring event:

scanning the compression group to determine whether the set of data has been compressed; and

determining whether any of the subsets in the set of data has been modified since a prior mirroring event; and

upon determining that the set of data has been compressed and that at least one of the subsets of the set of data has been modified, sending the set of data in its

entirety to a remote data storage system at a mirror site, for use in a mirror copy of the file.

52. An apparatus comprising:

means for receiving a file containing data; and

means for compressing at least part of the file to form a plurality of compression groups, each of the compression groups representing less than the entire file, each of the compression groups corresponding to an independently compressible group of data.

53. An apparatus as recited in claim 52, wherein said means for compressing at least part of the file comprises means for compressing the data represented by each of the compression groups independently.

54. An apparatus as recited in claim 53, further comprising means for determining suitability for compression independently for each of the compression groups.

55. An apparatus as recited in claim 52, wherein each of the compression groups has a plurality of pointers.

56. An apparatus as recited in claim 55, wherein at least one of the pointers in each of the compression groups points to compressed data, and wherein at least one other pointer in each of the compression groups is a predetermined value indicative that corresponding data has been compressed.

57. An apparatus as recited in claim 56, wherein the predetermined value further is indicative of the compression algorithm used to compress the data.

58. A processing system comprising:

means for receiving a set of data, the set of data having a first number of subsets;

means for creating a compression group corresponding to the set of data, the compression group including a plurality of entries, each entry containing a pointer to a corresponding one of the subsets;

means for compressing the set of data so that the set of data occupies a smaller number of the subsets than the first number; and

means for each of the subsets which does not contain compressed data after said compressing, storing a predetermined value in the corresponding entry of the compression group, the predetermined value being indicative that corresponding data is compressed.